

# Metadata of the chapter that will be visualized in SpringerLink

Book Title	Open Source Systems: Adoption and Impact	
Series Title		
Chapter Title	Open Source Software Ecosystems: Towards a Modelling Framework	
Copyright Year	2015	
Copyright HolderName	IFIP International Federation for Information Processing	
Corresponding Author	Family Name	<b>Franco-Bedoya</b>
	Particle	
	Given Name	<b>Oscar</b>
	Prefix	
	Suffix	
	Division	Group of Software and Service Engineering (GESSI)
	Organization	Universitat Politècnica de Catalunya
	Address	Barcelona, Spain
	Division	
	Organization	Universidad de Caldas
	Address	Manizales, Colombia
	Email	ohernan@essi.upc.edu
Abstract	<p>Open source software ecosystem modelling has emerged as an important research area in software engineering. Several models have been proposed to identify and analyse the complex relationships in OSS-ecosystems. However, there is a lack of formal models, methodologies, tool support, and standard notations for OSS-ecosystems. In this paper we propose a general framework for support the OSS-ecosystems modelling process. This framework will allow the representation, synthesis, analysis, evaluation, and evolution of OSS-ecosystems. Design science methodology is proposed to create several artefacts and investigating the suitability of these artefacts in the OSS-ecosystem context.</p>	
Keywords (separated by '-')	Modeling - Software ecosystem - Framework - Open source software	

# Open Source Software Ecosystems: Towards a Modelling Framework

Oscar Franco-Bedoya<sup>1,2</sup>✉

<sup>1</sup> Group of Software and Service Engineering (GESSI),  
Universitat Politècnica de Catalunya, Barcelona, Spain  
[ohernan@essi.upc.edu](mailto:ohernan@essi.upc.edu)

<http://www.essi.upc.edu/~gessi/>

<sup>2</sup> Universidad de Caldas, Manizales, Colombia

**Abstract.** Open source software ecosystem modelling has emerged as an important research area in software engineering. Several models have been proposed to identify and analyse the complex relationships in OSS-ecosystems. However, there is a lack of formal models, methodologies, tool support, and standard notations for OSS-ecosystems. In this paper we propose a general framework for support the OSS-ecosystems modelling process. This framework will allow the representation, synthesis, analysis, evaluation, and evolution of OSS-ecosystems. Design science methodology is proposed to create several artefacts and investigating the suitability of these artefacts in the OSS-ecosystem context.

**Keywords:** Modeling · Software ecosystem · Framework · Open source software

## 1 Introduction

Software ecosystem modelling (SEM) is one of the most studied areas in the software ecosystem domain [1]. However, currently there is no established modelling standard for software ecosystems. Nonetheless, there exists a need for modelling software ecosystems because:

- *Complexity.* Software ecosystems are among the most complex systems ever created by human [2], and models may help understanding them.
- *Traceability.* The software industry is constantly evolving and is currently undergoing rapid changes [3]. Models help in tracing the historic software ecosystem changes.
- *Communication.* Models help to represent the complex network of symbiotic relationships between entire social actors, open source communities and commercial software companies, etc. [4].

Ecosystem stakeholders need a common language to communicate their ideas. In the particular case of OSS Software Ecosystems, to our knowledge, there

are no contributions in the literature regarding how to model OSS-ecosystem support the representation and analysis of the specific relationships between OSS-ecosystem actors. The aim of the present work is to make a characterization of OSS-ecosystems, and to propose a specific framework for modelling OSS-ecosystems. This framework will be able to: represent, evaluate, evolve and analyse OSS-ecosystems.

## 2 Related Work

### 2.1 Software Ecosystems

The field of biological ecosystems has inspired several ecosystem domains. The business ecosystem (BECO) term was coined by Moore in 1996 [5]. The term digital business ecosystems (DBECO) was obtained by adding digital in front of business ecosystem [6]. the matured concept of DBECO was exposed by Briscoe that defined DBECO as distributed adaptive open socio-technical systems, with properties of self-organization, scalability and sustainability, inspired by natural ecosystems [7]. The term software ecosystem (SECO) has been coined by Messerschmitt and Szyperski to refer to *“a collection of software products that have some given degree of symbiotic relationships”* [8], [2]. However, in contrast to natural ecosystem, there is no common definition of software ecosystem. It can be defined and interpreted in different ways, depending on the point of view [2]. Some of the most accepted definitions of SECO are:

- A software ecosystem is *“a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them”* [1].
- A software ecosystem is *“a collection of software projects which are developed and evolve together in the same environment”* [9].
- A software ecosystem is *“a set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions”* [10].

### 2.2 Open Source Software Ecosystems

According to our knowledge of the literature, there are only a few definitions of OSS-ecosystem. These are provided by Wynn [11] and Hoving et al. [12]. Both authors define OSS-ecosystem based on the Iansiti and levien concept of business ecosystem (BECO) [13]. This mean that the shared market, organizations, capital, are the main actors that support the open source software community. In addition Jansen highlights the role of the developers in the OSS community and the freely nature of resources in its definition [12]. These concepts are a weak metaphor of natural ecosystem terms. However, the transfer of knowledge has essentially limited itself to a reuse of terms [2]. In order to support our proposal, In Table 1 we have integrated the different concepts of ecosystems:

**Table 1.** Open Software Ecosystem Definition

Definition	Sources	Examples
<i>An OSS-ecosystem is a SECO placed in a heterogeneous environment</i>	Iansity and Levin BECO	Other OSS ecosystems, commercial SECOs, Government, Market rules, etc.
<i>Its boundary is a set of niche players</i>	Jansen SECO	Partners, Re-sellers, Platform provider, etc.
<i>The keystone player is an OSS community around a set of projects in a common platform</i>	Lungu SECO	Contributors, passive users, data sources, etc.

OSS-ecosystems are complex artefacts that require a specific characterization in order to model its elements and relationships. The variety of ecosystem described above have common roles, (e.g. partners, users, developers, resellers, software products and services, etc.). However, in the OSS-ecosystems there are several conceptual and structural particularities, e.g., principles, community join process, goals, governance, legalities, among other concepts. A characterization conducted to evaluate similarity between commercial SECOs and OSS-ecosystems is described in detail in sections below.

### 2.3 Software Ecosystem Models

In the literature several specific models and meta-models have been proposed to identify and analyse the relationships between software ecosystems members. The work of Yu and Deng [3] and Lopez et al. [14] use i\* models to model the strategic dependencies between OSS-ecosystem actors. Boucharas et al., [15] and Jansen et al., [1] present the software ecosystem modelling (SEM) technique, which includes the product deployment context (PDC) and software supply networks (SSN) diagrams. A framework for sustainable software ecosystem management was discussed by Dhungana et al, [16]. Other authors like [17] and [18] represents the OSS-ecosystem actors and relationships using conceptual maps.

## 3 Research Methodology

We structure our research in terms of design science since it involves creating new artefacts and acquiring new knowledge, using an engineering cycle as main cycle and internal iterations with engineering activities and the empirical cycle Wieringa [19]. In our project the engineering cycle and the empirical cycle consist of five phases:

- *Problem investigation.* To investigate the nature of the problem we need to solve and to know which actions can help solve this problem.

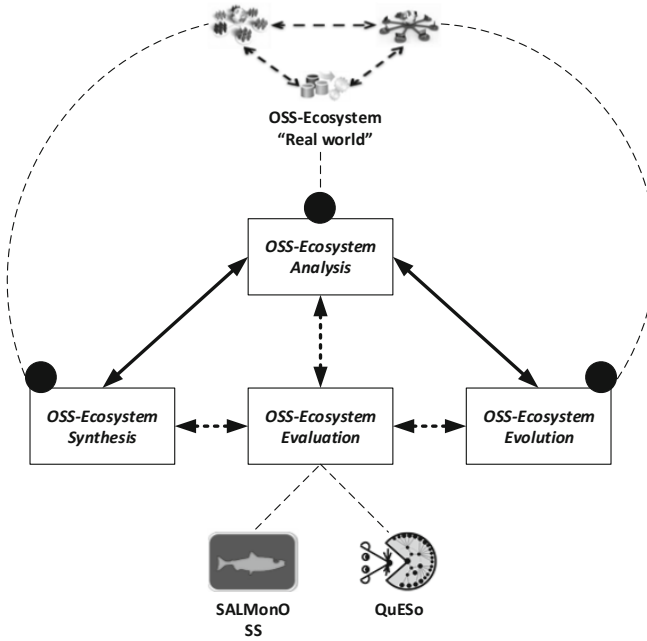
- *Treatment design.* To design the solution for the identified problem, in this phase it is necessary to evaluate several alternatives for each designed solution.
- *Design validation.* Before we construct the framework's components, we validate the designs to assure that the selected designs satisfy the criteria for the framework components.
- *Treatment implementation.* To realize the solution specification for the problem we will develop the framework, some components will be conceptual components and others software components.
- *Implementation evaluation.* Wieringa defines evaluation as the use of artefacts in context. We will validate the framework use case studies validation.

## 4 OSS-Ecosystem Modelling Framework

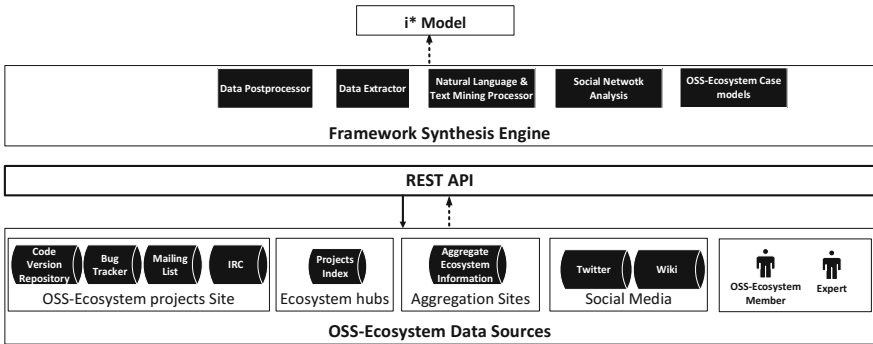
The ecosystem terminology defined by Mens [2] and Lopez [14] show that there are several differences between commercial SECOS and OSS-ecosystems. *Sustainability* in an OSS-ecosystem is related to the number of ecosystem community members [20]. On the other hand, in a commercial SECO it depends mainly on economic factors. The *adoption strategies* and the *adoption risks* derived by using OSS products in a company affect organizations business models [14]. In general, the most of the risks for adopting OSS components in an OSS-ecosystem are related to the licenses heterogeneity. *Governance* commercial software ecosystems are typically governed by a decision maker that decides how the ecosystem should evolve, while OSS-ecosystems often have a much more exible decisional structure [2]. The community is the *organizational* unit in OSS-ecosystems. In contrast, hierarchy structures are common in commercial SECO. An OSS-ecosystem modelling framework has to support: visualization, synthesis, analysis, evaluation and evolution of OSS-ecosystem models. In this section we provide a brief overview of the tools that support these activities. Our goal with this framework is to offer suggestions and ideas to researchers and practitioners in the field of OSS-ecosystem modelling. The framework that we propose is shown in Figure 1.

### 4.1 OSS-Ecosystem Model Synthesis

The purpose of this activity is to answer the question: *How is it possible to generate a specific OSS-ecosystem model only from OSS-ecosystem data sources?* Figure 2 shows a layered view of the components for this activity. At the bottom, there are several types of OSS-ecosystem data sources. Jansen defines three types [21]: project web sites, ecosystem hubs and aggregation sites. We added two other kinds of data sources: (1) social media sites such as Twitter, Facebook, etc. (2) strategic data from people related to the OSS-ecosystem obtain using specialized surveys. The OSS-ecosystem communities, typically provide open access to all data sources. The extraction of data is done with dedicated tools developed by the OSS-ecosystem researchers. Occasionally it is done by the use



**Fig. 1.** OSS-ecosystem modelling framework



**Fig. 2.** OSS-Ecosystem Model Synthesis

of specialized tools. E.g. Spanish LibreSoft group provides FLOSSMetrics to extract data from repositories and then they are stored in a set of databases [22]. However, most of these tools are not reusable in other experiments, even in the same OSS-ecosystem. The availability of the data depends of the OSS-ecosystem. Because of this, we propose to define an extensible REST API. This is a set of web

services to be implemented by each OSS-ecosystem community. This interface would allow obtain information related to the OSS-ecosystem.

Our aim is to motivate the development of this API by the OSS-communities providing them with a framework for modelling, analysing and monitoring their own OSS-ecosystem. The framework synthesis engine has two extraction components, similar to Goemmine et al. [22]. Moreover, It uses the OSS ontology defined by López et al. [14], social network analysis (SNA), self-modelling techniques and predefined OSS-ecosystem models to identify the OSS-ecosystem actors and relationships in a specific OSS-ecosystem (e.g. Eclipse, Gnome, etc. . Finally, the synthesis components will generate an i\* OSS-ecosystem model.

## 4.2 OSS-Ecosystem Model Valuation

This component will enable the monitoring of the OSS-ecosystem health. To prove the feasibility of the approach we propose develop this component based on an existing technologies named SALMonOSS [23] and QuESo [20] developed in our research group.

- *SALMonOSS*. The general idea is to adopt principles and methods from the service oriented computing field (SOC). Particularly, we propose to adapt the concepts of quality service and service level agreement, and propose to reuse the existing body of knowledge and techniques from SOC monitoring. Figure 3 shows the OSS-ecosystem evaluator. SALMonOSS is an OSS-ecosystem health monitor component able to: (1) monitor a list of ecosystem health indicators along time (2) link the gathered values with client's needs by operationalization of conditions in software ecosystem agreements (SELAs) and (3) engineer a portfolio of methods and techniques that supports OSS ecosystems (e.g. OSS selection, proactive adaptation, etc.).
- *QuESo*. QuESo is a quality model for assessing the quality of OSS ecosystems. QuESo have been organized in three dimensions: (1) those that relate to the platform around which the ecosystem is built, (2) those that relate to the community of the OSS-ecosystem and (3) those that are related to the ecosystem as a network of interrelated elements, such as projects or companies. We are using QuESo to define the key health indicators (KHIs) to be monitored by SALMonOSS. the SELAs and the software ecosystem level fulfilment (SELF are composed by KHIs).

## 4.3 OSS-Ecosystem Model Analysis

If there is a defined OSS-ecosystem model, what type of data or functionality should be changed in the OSS-ecosystem to satisfy the proposal model? Figure 4 shows the main components for this activity. To answer this question we will use: an OSS-ecosystem ontology, the expert system engine and the case base reasoning (CBR). The rules defined in the ontology allow reasoning about the class instances and their relationships obtained from the OSS-ecosystem data sources.

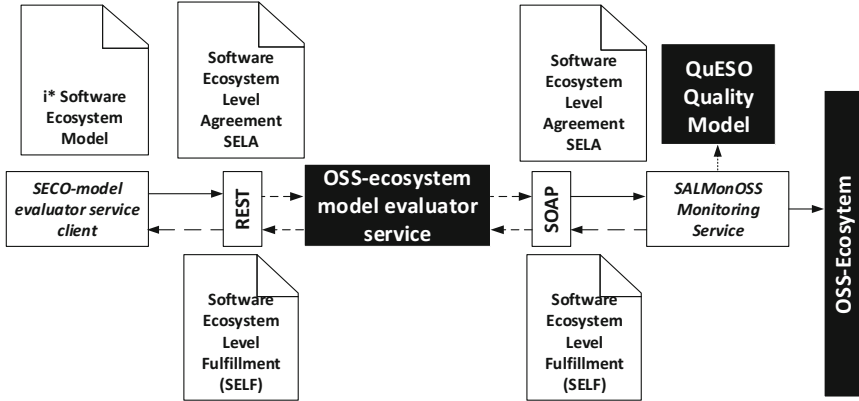


Fig. 3. OSS-Ecosystem Model Evaluation

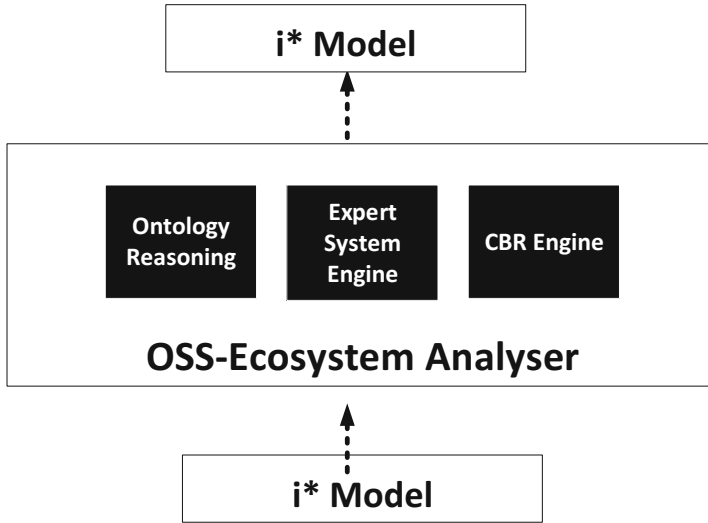


Fig. 4. OSS-Ecosystem Model Analysis

The expert system engine will be used to register the knowledge obtained from the software ecosystem experts about specific OSS-ecosystem models. Finally, we will use CBR reasoning to select strategies for propose possible changes in the initial OSS-ecosystem model defined.

#### 4.4 OSS-Ecosystem Model Evolution

The OSS-ecosystem are dynamics and complex artefacts. Similar to the natural ecosystem the actors, roles, dependencies, resources, relationships, etc., changed



frequently over time. The question is: Once a specific OSS-ecosystem model has been created, how its continuous evolution along time can be done?. The software ecosystem evolution is stored in the OSS-ecosystem repositories. Since the software environment involves human beings (developers and users). This makes it possible, in principle, to interact with them in order to find out how and why a software project has evolved over time, and making it easier to alter the way in which the ecosystem will evolve in the future.[2]. With the tools defined in our framework and with the information stored in the OSS-ecosystem repositories, We will be able to visualize the changes in the OSS-ecosystem model structure, interactions, health, releases, resources, etc., all this from a social technical perspective. E.g. OSS-communities, legalities, partners, platform, technologies and projects.

## 5 Conclusions and Future Work

In this paper we have presented a general framework for representation, synthesis, analysis, evaluation and evolution of OSS-ecosystems. We believe that ecosystem modelling is a promising research direction and we plan to continue working on it. Our focus is would be defining methodologies, languages, formal syntax and semantic rules for modelling software ecosystem based on the models and metamodels described in the literature. In a first stage, we are working in the QuESo quality model validation and its integration with SALMonOSS framework.

**Acknowledgments.** This work is a result of the RISCOSS project, funded by the EC 7th Framework Programme FP7/2007-2013 under the agreement number 318249. We would also like to thank the contribution of EOSSAC project, founded by the Ministry of Economy and Competitiveness of the Spanish government (TIN2013-44641-P).

## References

1. Jansen, S., Brinkkemper, S., Finkelstein, A.: Business network management as a survival strategy: A tale of two software ecosystems. In: Proceedings of the 1st Workshop on Software Ecosystems, CEUR-WS, pp. 34–48 (2009)
2. Mens, T., Claes, M., Grosjean, P., Serebrenik, A.: Studying evolving software ecosystems based on ecological models, pp. 297–326. Springer, Heidelberg (2014)
3. Yu, E., Deng, S.: Understanding software ecosystems: A strategic modeling approach. In: Proceedings of the 3th Workshop on Software Ecosystems, IWSECO, pp. 65–76 (2011)
4. Yamakami, T.: A three-layer view model of oss: Toward understanding of diversity of oss. In: Proceedings of the 13th International Conference on Advanced Communication Technology, ICACT, pp. 1190–1194 (2011)
5. Moore, J.F.: Predators and prey: a new ecology of competition. *Harvard Business Review* **71**, 75–83 (1993)
6. Stanley, J., Briscoe, G.: The ABC of digital business ecosystems. *Computer, Media and Telecommunications Law* **15**, 1–24 (2010) P28

7. Briscoe, G.: Digital Ecosystems. PhD thesis, Imperial College London (2009)
8. Messerschmitt, D.G., Szyperski, C.: Software Ecosystem: Understanding an Indispensable Technology and Industry. MIT Press Books, vol. 1. The MIT Press (2005)
9. Lungu, M., Malnati, J., Lanza, M.: Visualizing gnome with the small project observatory. In: Proceedings of the 6th IEEE International Working Conference Mining Software Repositories, MSR, pp. 103–106 (2009)
10. Bosch, J.: From software product lines to software ecosystems. In: Proceedings of the 13th International Software Product Line Conference, SPLC 2009, pp. 111–119. Carnegie Mellon University, Pittsburgh (2009)
11. Wynn Jr, D., Boudreau, M.C., Watson, R.: Assessing the Health of an Open Source Ecosystem. GI Publishing, New York (2008)
12. Hoving, R., Slot, G., Jansen, S.: Python: Characteristics identification of a free open source software ecosystem. In: Proceedings of the 7th IEEE International Conference on Digital Ecosystems and Technologies, DEST, pp. 13–18 (2013)
13. Iansiti, M., Levien, R.: The keystone advantage: what the new dynamics of business ecosystems mean for strategy, innovation, and sustainability. Harvard Business Press (2004)
14. López, L., Costal, D., Ayala, C.P., Franch, X., Glott, R., Haaland, K.: Modelling and applying OSS adoption strategies. In: Yu, E., Dobbie, G., Jarke, M., Purao, S. (eds.) ER 2014. LNCS, vol. 8824, pp. 349–362. Springer, Heidelberg (2014)
15. Boucharas, V., Jansen, S., Brinkkemper, S.: Formalizing software ecosystem modeling. In: Proceedings of the 1st International Workshop on Open Component Ecosystems, IWOCE, pp. 41–50. ACM, New York (2009)
16. Dhungana, D., Groher, I., Schludermann, E., Biffl, S.: Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In: Proceedings of the 4th ECSA, pp. 96–102. ACM (2010) P48
17. Morgan, L., Feller, J., Finnegan, P.: Exploring value networks. *Eur J Inf Syst* **22**, 569–588 (2013)
18. Mattmann, C.A., Downs, R.R., Ramirez, P.M., Goodale, C., Hart, A.F.: Developing an open source strategy for NASA earth science data systems. In: Proceedings of the 13th IRI, pp. 687–693. IEEE (2012) P15
19. Wieringa, R.: Design science as nested problem solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, pp. 8:1–8:12. ACM, New York (2009)
20. Franco-Bedoya, O., Ameller, D., Costal, D., Franch, X.: Queso: A quality model for open source software ecosystems. In: Proceedings of the 9th International Conference on Software Engineering and Applications, ICSEEA, pp. 209–221 (2014)
21. Jansen, S.: Measuring the health of open source software ecosystems: Beyond the scope of project health. *Information and Software Technology* **56**, 1508–1519 (2014) Special issue on Software Ecosystems P68
22. Goeminne, M., Mens, T.: Analyzing ecosystems for open source software developer communities. In: *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, pp. 247–275. Edward Elgar Publishing (2013) P57
23. Oriol, M., Franco-Bedoya, O., Franch, X., Marco, J.: Assessing open source communities' health using service oriented computing concepts. In: Proceedings of the 8th International Conference on Research Challenges in Information Science (RCIS), pp. 1–6. IEEE (2014)